

# *Performance Measures for Geometric Fitting in the NIST Algorithm Testing and Evaluation Program for Coordinate Measurement Systems*

---

Volume 100

Number 5

September–October 1995

---

**Theodore H. Hopp and  
Mark S. Levenson**

National Institute of Standards and  
Technology,  
Gaithersburg, MD 20899-0001

The Algorithm Testing and Evaluation Program for Coordinate Measurement Systems (ATEP-CMS) is a Special Test Service offered under the NIST Calibration Program. ATEP-CMS evaluates the performance of geometric fitting software used in coordinate measurement systems. It is a Special Test because it is a new type of NIST service, experimental in nature and unsupported by historical data. This report documents and explains the rationale of the performance measures used in ATEP-CMS

and analyzes the uncertainties of those measures.

**Key words:** algorithm testing service; computational metrology; coordinate measurement systems; dimensional metrology; geometric fitting; measurement uncertainty; mechanical tolerancing; orthogonal distance regression; software uncertainty.

**Accepted:** June 8, 1995

---

## 1. Introduction

This report is concerned with the performance of software used in coordinate measurement systems (CMSs) to evaluate the geometric characteristics of manufactured parts. The particular performance characteristics of interest are those that impact the uncertainty of measurement results produced by CMSs.

Inspection planners need quantitative measures of performance to develop uncertainty budgets and to evaluate the quality of measurement results. In support of this need, NIST has recently established a Special Test Service, the Algorithm Testing and Evaluation Program for Coordinate Measurement Systems (ATEP-CMS), to measure the performance of geometric fitting software used in CMSs [1]. This report documents and explains the performance measures used in ATEP-CMS.

Geometric fitting is the process of computing the representation parameters of a geometric element that in some sense best represents a set of point coordinate data. This representative geometric element is called the

*substitute geometry* for the data points. A manufactured hole, for instance, is usually not perfectly cylindrical because the process that produced it can never be totally perfect. Inspection of the hole might involve measuring the coordinates of selected points on the surface of the hole, fitting a cylinder to the measured points to minimize the sum of squares of the orthogonal distances from the points to the cylinder, and comparing the position, orientation, and size of the fitted cylinder to the dimensions and tolerances of the part specification.

Many factors affect the accuracy of inspection procedures.<sup>1</sup> This report focuses on one of these factors: how close the computed fit is to the intended, mathematically defined, substitute geometry. The performance

<sup>1</sup> ATEP-CMS—part of a larger field of endeavor called *computational metrology* [2]—addresses the *fitting software* as a source of uncertainty due to possible computational errors. It does not address the propagation of point coordinate uncertainty through the geometric fitting computations, how well the points represent the surface, or whether the intended software function is the most appropriate for the task.

measures used in ATEP-CMS quantify how well the software computes substitute geometries over a range of inspection problems.

The next section provides background information on the need for algorithm testing, how ATEP-CMS works, and criteria for performance measures. Section 3 presents details of the test methods used in ATEP-CMS for various types of geometric elements. Section 4 discusses a statistical interpretation of algorithm performance and explains the method used in ATEP-CMS to summarize and interpret the test results. Section 5 presents an uncertainty analysis of ATEP-CMS results.

## 2. Background

The drive to develop methods for testing CMS algorithms began in Germany in the early 1980s, when it became apparent that the software supplied with commercial coordinate measurement machines (CMMs) varied widely in quality [3, 4]. In the United States, the American Society of Mechanical Engineers established a task force on CMM software that is now designated B89.4.10. Work intensified in 1988 following widespread notification within the defense industries of serious problems with certain commercial CMM software [5]. A U.S. standard on performance evaluation of CMS software is now in draft form and is expected to be issued for public review in early 1996 [6]. NIST's ATEP-CMS service [1] supports this emerging standard.

### 2.1 What is Evaluated by ATEP

When discussing software, the term “testing” is used in many different ways, and it is important to understand the sense in which it is used within ATEP-CMS. It is specifically *not* used in the sense of software engineering, where software testing is aimed at supporting debugging or maintenance and typically involves code structure analysis, walk-throughs, and similar activities. Within ATEP-CMS, testing is strictly limited to black-box testing of how the actual behavior of the software compares with its intended behavior. Moreover, ATEP-CMS restricts itself to measuring the accuracy of reported results. Other performance characteristics—such as memory requirements, computing time, ease of use, and other factors—are not considered. In sum, ATEP-CMS ignores the fact that the software under test is software; the ATEP-CMS process would be unchanged if the fitting methods were implemented with an analog computer, as a digital filter circuit, or even mechanically [7].

ATEP-CMS does not address whether the intended behavior of the software is *appropriate* for a particular measurement task. That issue is not one of performance, but of whether a particular data analysis method is the right tool for the job. Such considerations are beyond the scope of a performance testing program.

ATEP-CMS currently supports testing of orthogonal distance regression fitting<sup>2</sup> for seven geometry types: line, circle, plane, sphere, cylinder, cone, and torus. (Lines and circles are fit to three-dimensional point coordinates which need not be coplanar.) The reader is referred to Ref. [9] for information on how to use ATEM-CMS and the procedures used by NIST for conducting a test. Section 3 below describes the specific difference parameters used in ATEP-CMS for each geometry type.

### 2.2 Criteria for Performance Measures

In developing performance measures, the question naturally arises as to how one might choose one measure over another. One approach is to seek a reliable method of certifying software as “good” or “bad.” (This is the approach taken, for instance, by the German testing program [3].)

The approach taken in ATEP-CMS is somewhat different. We start with the observation that software is not mathematically perfect. (If nothing else, the end results must be represented in finite precision.) So rather than developing a pass/fail criterion, ATEP-CMS quantifies the expected performance numerically. The goal is to develop numerical measures of performance that provide to a user of the software, information necessary to decide whether the software is adequate for a particular application. To this end, we have used the following criteria in developing ATEP-CMS performance measures:

- each measure should be directly related to inspection tasks;
- each measure should combine like a standard deviation with other sources of uncertainty in a CMS;

<sup>2</sup> Orthogonal distance regression is commonly called “least squares” by manufacturing practitioners. Statisticians consider these two terms to be quite different. Least squares measures deviations from the surface along a particular direction in a coordinate system. Orthogonal distance regression (which is also called “errors in variables” regression or, for linear fits, “total least squares”) measures deviations perpendicular to the surface. Unfortunately, some of the literature (e.g., Ref. [8]) further confuses the terminology by using the term “orthogonal distance regression” for one specific algorithm that solves the orthogonal distance regression problem.

- each measure should have reasonable probability and coverage interpretations;
- an estimate of uncertainty should be derivable for each measure.

The first three criteria directly support the goal of using the performance measures to establish uncertainty budgets for CMSs. The last criterion reflects the view that an assessment of software uncertainty is itself a measurement. By NIST [10] and international [11] guidelines, a quantitative statement of uncertainty should be part of any complete measurement result. The last criterion says that the ATEP-CMS performance measures must be amenable to such a treatment.

### 3. ATEP-CMS Test Methods

To start a software test, NIST generates a collection of data sets for each geometry type, consisting of three-dimensional point coordinates together with some book-keeping information. NIST also generates a fit for each data set, called the *reference fit* for that data set. Within ATEP-CMS, the reference fits are generated by fitting the data sets using NIST-developed fitting algorithms.<sup>3</sup> (It is also possible to start with the desired fits and generate data sets having those fits as answers [13].)

The data sets are then sent to the software under test, which generates a *test fit* for each data set. The test fits, represented by a set of parameter values in a standard format, are returned to NIST for evaluation with respect to the corresponding reference fits.

This section describes how differences between a test fit and the corresponding reference fit are evaluated in ATEP-CMS. Differences between each pair of fits are represented by a set of *difference parameters*. Each geometry type gives rise to its own set of difference parameters. The relationship between the difference parameters and tolerance applications is also discussed.

#### 3.1 General Procedure for a Single Fit

Methods for comparing one fit to a reference are described in the draft B89.4.10 standard [6]. The procedure for all geometry types follows a common pattern.

First, the test fit is bounded by projecting the data points onto the geometry. (The way this is done depends

<sup>3</sup> A small disclaimer is in order here. The use of the term “reference fit” is not meant to imply that the ATEP-CMS fitting algorithms are “standard algorithms” for geometric fitting problems. For instance, the algorithm used in ATEP-CMS for nonlinear orthogonal distance regression (see Sec. 5.1.3) was chosen for its efficient use of memory, but its use of normal equations in the iteration are a known weakness. Algorithms with superior numerical properties are widely available (see, e.g., Refs. [12, 8]) and may be used by ATEP-CMS in the future.

on the geometry type, and will be described below. Also, bounding does not apply to circles, spheres, or tori, which are naturally bounded.) This is done because tolerancing standards (e.g., Ref. [14]) specify that tolerances are to be evaluated over the full extent of the associated geometric features. ATEP-CMS assumes that the data points represent the features.

Once the test fit is bounded, geometric differences between the test fit and the reference fit are determined. Again, the specifics depend on the geometry type. In general, however, the differences are designed either to directly reflect a tolerancing application or to provide diagnostic information about the software under test. The difference parameters depend only on the geometry represented by the fit parameters, and not on the parameterization of that geometry.

Finally we note that in many cases the difference parameters are not symmetric. This is a result of the asymmetric treatment of the test and reference fit (the test fit is bounded; the reference fit is not). Thus, reversing the roles of test and reference fit will generally change the difference parameter values.

#### 3.2 Application to Specific Geometry Types

The remainder of this section discusses the application of the general procedure to the seven geometry types currently supported by ATEP-CMS: lines, circles, planes, spheres, cylinders, cones, and tori. For each geometry type, we describe the fitting objective, typical uses for such fits in inspection, and the difference parameters computed within ATEP-CMS.

**3.2.1 Lines** Line fitting minimizes the sum of squares of orthogonal distances from the points to the fit line. Line fitting is commonly used to check straightness and to establish an axis from the centers of circular cross sections. In the latter case, the axis may be checked for location or orientation or used as a coordinate datum axis.

Within ATEP-CMS, two parameters representing differences between the lines are computed. (Henceforth, we will call such parameters, *difference parameters*.) One difference parameter is the angle between the test and reference lines. This directly relates to the use of the fit as a datum axis. To define the other parameter, the test fit line is first trimmed to a segment by the orthogonal projection of the data points. Then the second parameter is the maximum orthogonal distance from the segment endpoints to the reference line. This is a measure of separation between the lines and directly relates to the assessment of straightness, position, and orientation. Both parameters are inherently non-negative quantities. (Henceforth, quantities that are inherently nonnegative will be referred to as *magnitudes*.)

**3.2.2 Circles** Circle fitting is a two-step process. First, the plane of a circle is found by fitting a plane to the data (as described below). Then, the points are projected into the plane and an orthogonal distance regression circle is fit to the projected points. Note that this process generally results in a different circle than would be found by using the three-dimensional distances from the points to the circle.

Circle fitting is commonly used to check roundness, runout, cross-sectional size, the straightness of the median line of a hole or shaft, and sometimes position. It is also sometimes used as one step in finding an axis. The purpose of the two-step fitting process is that, in practice, the deviations of the points from a plane usually result from measurement error only, while the deviations from a circle in the plane also include the effects of form deviation of the part.

Within ATEP-CMS, three difference parameters are computed. One is the angle between the planes of the test and reference circles. This is primarily a diagnostic measure<sup>4</sup> that can explain other differences between the fits. The second parameter is the Euclidean distance between the circle centers, measured in three dimensions. This relates directly to assessment of roundness, position, and establishing an axis. The third parameter is the difference between the test and reference circle radii. This relates directly to the assessment of size. The first two parameters are magnitudes, while the last parameter is a signed quantity, and is negative whenever the test fit radius is smaller than the reference fit radius.

**3.2.3 Planes** Plane fitting minimizes the sum of squares of orthogonal distances from the points to the fit plane. Plane fitting is commonly used to establish a coordinate datum plane and to check flatness, position, and orientation.

Within ATEP-CMS, two difference parameters are computed. One parameter is the angle between the test and reference planes. This directly relates to the use of the fit as a datum plane. To define the second parameter, the data points are orthogonally projected onto the test fit plane. The convex hull of the projected points in the plane forms a planar patch. The second parameter is the maximum separation of the planar patch measured orthogonally from the reference plane. This measure of separation directly relates to the assessment of flatness, position, and orientation. Both parameters are magnitudes.

**3.2.4 Spheres** Sphere fitting minimizes the sum of squares of orthogonal distances from the points to the fit sphere. Sphere fitting is commonly used to check

form, size, and location, and occasionally to locate the origin of a coordinate system.

Within ATEP-CMS, two difference parameters are computed. The first parameter is the Euclidean distance between the sphere centers. This relates directly to assessment of form, position, and establishing an origin. The second parameter is the difference between the test and reference sphere radii. This relates directly to the assessment of size. The first parameter is a magnitude, while the second is a signed quantity, and is negative whenever the test fit radius is smaller than the reference fit radius.

**3.2.5 Cylinders** Cylinder fitting minimizes the sum of squares of orthogonal distances from the points to the fit cylinder. Cylinder fitting is commonly used to check cylindricity and size, and to establish an axis. In the latter case, the axis is used to check position and orientation and to establish a coordinate datum.

Within ATEP-CMS, three difference parameters are computed. One parameter is the angle between the test and reference cylinder axes. This directly relates to the use of the fit as a datum axis. To define the second parameter, the test fit axis is first trimmed to a segment by the perpendicular projection of the data points. Then the second parameter is the maximum perpendicular distance from the segment endpoints to the reference cylinder axis. This is a measure of separation between the axes and directly relates to the assessment of cylindricity, position, and orientation. The third parameter is the difference between the test and reference cylinder radii. This relates directly to the assessment of size. The first two parameters are magnitudes, while the third is a signed quantity, and is negative whenever the test fit radius is smaller than the reference fit radius.

**3.2.6 Cones** Cone fitting minimizes the sum of squares of orthogonal distances from the points to the cone surface. Cone fitting is commonly used to check profile and conical taper.

Within ATEP-CMS, four difference parameters are computed. One is the angle between the test and reference cone axes. This directly relates to the use of the fit for checking profile. To define the next parameter, the test fit axis is first trimmed to a segment by first projecting the data points perpendicularly onto the cone surface and then orthogonally projecting the projections onto the cone axis. Then the second parameter is the maximum perpendicular distance from the segment endpoints to the reference cone axis. This is a measure of separation between the axes and relates to the assessment of profile. To define the third parameter, the reference cone axis is bounded using the same process as was used for the test fit. For each fit, the perpendicular distance from the midpoint of the axis segment to the corresponding cone surface is a measure of location of

<sup>4</sup> For instance, it can be used to detect if something other than the two-step fitting process was used by the software under test.

the cone along its axis. The third parameter is the difference between the location of the test fit cone and the location of the reference fit cone along their respective axes. This is primarily diagnostic, and relates to the assessment of profile. The last (fourth) parameter is the difference between the test and reference cone half-angles. This relates directly to the assessment of conical taper. The first two parameters are magnitudes, while the last two are signed quantities, and are negative whenever the test fit cone parameter is smaller than that of the reference fit.

**3.2.7 Tori** Torus fitting minimizes the sum of squares of perpendicular distances from the points to the torus surface. Torus fitting is commonly used to check major and minor radii and profile. Occasionally torus fitting is used to establish a datum axis or plane.

Within ATEP-CMS, four difference parameters are computed. One is the angle between the test and reference torus planes. This directly relates to the use of the fit for checking profile and as a datum plane or axis. The second is the Euclidean distance between the torus centers. This also relates to the assessment of profile. The third and fourth parameters are the difference between the test and reference fit major and minor radii. These relate directly to the assessment of major and minor radius tolerances. The first two parameters are magnitudes, while the last two are signed quantities, and are negative whenever the test fit torus parameter is smaller than that of the reference fit.

## 4. Data Interpretation

The previous section described how each data set generates a set of difference parameters representing the differences between the test fit to the data set and the corresponding reference fit. In this section we discuss how these difference parameters are summarized into performance measures. For each difference parameter, we will define an associated performance measure summarizing the difference parameter values for all the data sets.

We consider the set of difference parameter values associated with each data set to be one (random) sample of software performance from a theoretical underlying population of inspection problems. We will use a statistical approach to the interpretation of the observed sample values. In Sec. 4.1 we derive the population characteristic we wish to use as a performance measure. In Sec. 4.2 we derive a practical estimator for that measure.

### 4.1 Statistical Model of Algorithm Behavior

We deal with each geometry type separately, and consider each performance measure (and associated difference parameters) independently. For simplicity,

consider a single performance parameter for a given geometry type, as described in Sec. 3. Each data set represents an inspection problem drawn at random from a theoretical underlying population of inspection problems. We call the substitute geometry defined by the mathematical objective function the *true fit* to the data set. We assume the true fit cannot be computed exactly. For the  $i$ th data set, we now define three difference parameter values:

$t_i$ —the difference between the test fit and the true fit to the data set;  $t_i$  is unknown and unobservable.

$r_i$ —the difference between the reference fit and the true fit to the data set;  $r_i$  is unobservable, but it can be bounded using numerical analysis theory (see Sec. 5).

$p_i$ —the difference between the test fit and the reference fit to the data set;  $p_i$  is the calculated difference parameter for the  $i$ th data set.

These quantities are random variables since they are functions of the data set, which is a random sample. As random variables, we can define various expectations. For the test fit quantities  $t_i$  we define the mean

$$\mu_T = E(t_i);$$

the standard deviation

$$\sigma_T = \sqrt{E((t_i - \mu_T)^2)};$$

and the root-mean square error (rmse)

$$\gamma_T = \sqrt{E(t_i^2)},$$

where  $E( )$  denotes the average over all the data sets—that is, the expected value taken over the theoretical population of inspection problems. We define analogous quantities for  $r_i$ :  $\mu_R$ ,  $\sigma_R$ , and  $\gamma_R$ ; and for  $p_i$ :  $\mu_P$ ,  $\sigma_P$ , and  $\gamma_P$ .

If the  $t_i$  are normally distributed, then one can make statements of coverage like

$$\Pr \{ |t_i - \mu_T| \leq 2\sigma_T \} \approx 0.95. \quad (1)$$

This approach has two shortcomings. First, two parameters,  $\mu_T$  and  $\sigma_T$ , are needed to summarize the test; for simplicity, we would like a single measure of performance. Second, it is difficult to estimate  $\mu_T$  and  $\sigma_T$ , because we do not know the true fit.

If  $\mu_T$  is less than half of  $\sigma_T$ , then the rmse provides similar coverage.<sup>5</sup>

$$\Pr \{ |t_i| \leq 2\gamma_T : \mu_T < \sigma_T/2 \} \approx 0.95.$$

If  $\mu_T > \sigma_T/2$ , the actual coverage is greater (the software performance is better) than that suggested by a measure based on  $\gamma_T$ .

The above coverage interpretations are valid for normal distributions. We believe that for the typical distributions that will arise, the coverage error will be on the conservative side since the tails are likely to be smaller than those of a normal distribution.<sup>6</sup>

ATEP-CMS uses  $\gamma_T$  as the theoretical performance measure of the software under test. Using the rmse overcomes one shortcoming of Eq. (1)—we have reduced the performance indicator to a single number. We have also replaced the problem of estimating  $\mu_T$  and  $\sigma_T$  with the problem of estimating  $\gamma_T$ . We consider this next.

## 4.2 Definition of Practical Performance Measures

In this section, we will describe two methods of estimating  $\gamma_T$ . The first follows directly from the definition of  $\gamma_T$ . The second is a simplified estimate. ATEP-CMS uses the second method, for reasons discussed at the end of this section.

We observe from the definitions that  $t_i = p_i + r_i$  for signed parameters, and  $t_i \leq p_i + r_i$  for magnitudes. Then a simple substitution yields

$$\begin{aligned} \gamma_T^2 &= E(t_i^2) \\ &\leq E(p_i^2) + E(r_i^2) + 2E(p_i r_i) \\ &\leq \gamma_R^2 + \gamma_P^2 + 2\gamma_P \gamma_R. \end{aligned}$$

Thus, we can estimate an upper bound for  $\gamma_T$  if we can estimate  $\gamma_R$  and  $\gamma_P$ . The upper bound can then be used as an estimate for  $\gamma_T$ .

To estimate  $\gamma_R$ , we need an estimate of the difference of each reference fit from the true fit. Methods for doing this will be discussed in detail in Sec. 5. For now, assume that the value of  $r_i$  can be estimated as  $u_i$ . Then an estimate<sup>7</sup>  $\hat{\gamma}_R$  of  $\gamma_R$  is the positive square root of

$$\hat{\gamma}_R^2 = \frac{1}{n} \sum_{i=1}^n u_i^2,$$

<sup>5</sup> See, e.g., Ref. [15, Chap. 2].

<sup>6</sup> This assumption will be tested using data collected during future operation of ATEP-CMS.

<sup>7</sup> Throughout this paper, a carat appearing over a quantity denotes an estimated value.

where the sum is over all  $n$  data sets. (In this paper, all summations are from  $i = 1$  to  $n$  unless otherwise indicated. Henceforth, we will use the summation sign alone with the index variable and limits understood.) Similarly, we estimate  $\gamma_P$  by the positive square root of

$$\hat{\gamma}_P^2 = \frac{1}{n} \sum p_i^2.$$

This first estimation method includes a term in the performance measure that represents the estimated difference between the reference and true fits.

A second method of estimating  $\gamma_T$  is simply  $\hat{\gamma}_T = \hat{\gamma}_P$ , estimating  $\gamma_P$  as before. This method uses the reference fit as an estimate of the true fit. Unlike the first approach, the differences between the reference and true fits are not incorporated in the performance measure of the software under test. However, they will contribute to the uncertainty of the performance measures (discussed in Sec. 5).

An argument in support of the first method of estimation is that the second, simpler method may underestimate the true quantity. If the performance measures are to be used to establish uncertainty budgets for a CMS, use of the second method makes it critical to analyze the sensitivity of the budget to the uncertainties of the ATEP-CMS results. The first method may have smaller uncertainties.

On the other hand, there are three reasons why the simpler estimate is preferable. First, the software under test should not be penalized simply because the reference software has uncertain performance. Second, the first method may overemphasize the role of the reference software, and may be unduly pessimistic (since it is an upper bound). Finally, a sensitivity study is a proper precaution, and should be recommended anyway.

For these reasons, ATEP-CMS uses the second method—the observed sample rmse—to estimate the performance of the software under test. That is, for each performance measure in ATEP-CMS,

$$\hat{\gamma}_T = \hat{\gamma}_P = \sqrt{\frac{1}{n} \sum p_i^2}.$$

## 5. Uncertainty Analysis

This section addresses the uncertainty of the ATEP-CMS estimates of software performance. Recall that we have defined different performance measures for each geometry type, and that each performance measure is estimated from the corresponding set of difference parameters for the fits. We will show in this section that the uncertainty of each estimate arises mainly from two

sources: (1) the uncertainty of the individual difference parameters due to the uncertainty of the reference fit and (2) the uncertainty arising from having sampled the population of inspection problems.

In Sec. 5.1 we will first examine the uncertainties of the reference fits themselves. In Sec. 5.2 we then examine how these uncertainties propagate to the difference parameters. Finally, in Sec. 5.3 we discuss how the difference parameter uncertainties combine with the sampling uncertainty to yield an overall uncertainty for each performance measure.

### 5.1 Uncertainty of Reference Fits

The uncertainties of the difference parameters for a single test fit arise primarily from the uncertainty due to the presence of (unknown) numerical errors in the reference fit.<sup>8</sup> That is, the reference fit may not correspond exactly to the true fit. A detailed discussion of the NIST fitting algorithms and their uncertainties will be presented in a subsequent report. Here, we provide a summary of the pertinent results.

Our approach is to develop bounds for the numerical errors, assume the errors are uniformly distributed between the bounds, and combine the standard deviations of the errors into the uncertainty of the reference fits. The results presented in this section and the next are specific to the particular algorithms currently used in ATEP-CMS. It is possible that the algorithms may change in the future [16], in which case these results would no longer hold. (The performance measures would not change, however.)

**5.1.1 Centroid Coordinates** For all fits, the data are first translated to be centered at the origin. One source of uncertainty is the possible rounding error in computing the centroid. The ATS algorithms use the Kahan summation algorithm [17] with a C-language long double accumulator (80 bit IEEE extended real format). The error in the coordinates of the computed centroid can be bounded in terms of the unit roundoff  $\varepsilon_{80}$  for the accumulator (about  $1.084 \times 10^{-19}$ ). For instance, if  $x_c$  is the computed  $x$  coordinate of the centroid (the average of the coordinates  $x_i$ ,  $i = 1, \dots, m$ ) and  $x_0$  is the true coordinate, then the following result holds:

$$\begin{aligned} x_c &= \frac{1}{m} \sum_{i=1}^m x_i (1 + \delta_i) + O(10^{-38}) \sum_{i=1}^m |x_i| \\ &\approx x_0 + \frac{1}{m} \sum_{i=1}^m x_i \delta_i, \end{aligned}$$

where  $|\delta_i| \leq 2\varepsilon_{80} \approx 2.17 \times 10^{-19}$  and in the second line we have ignored the term in  $O(10^{-38})$ . Similar results

<sup>8</sup> Other errors, such as floating-point roundoff during evaluation of the difference parameters, are considered small enough to be ignored.

hold for the  $y$  and  $z$  coordinates.

To conform to NIST policy on uncertainty statements, this must be converted to an uncertainty by assuming some distribution on the possible values of  $\delta_i$ . We will assume that the  $\delta_i$  are independent and uniformly distributed within the bounds [18]. Then the  $i$ th error term  $x_i \delta_i$  is a random variable with mean of zero and standard deviation  $2|x_i| \varepsilon_{80}/\sqrt{3}$ . So, by the Central Limit Theorem [19], the  $x$  coordinate of the computed centroid of  $m$  data points is approximately normally distributed with mean  $x_0$  and standard deviation of

$$\frac{1.25 \times 10^{-19} \sqrt{\sum x_i^2}}{m}$$

Similar results hold for the  $y$  and  $z$  coordinates. For simplicity, the uncertainty of the centroid is modeled in ATEP-CMS by an isotropic, trivariate normal distribution using a standard deviation of

$$u_0 = 1.25 \times 10^{-19} \sigma_{\max}/m,$$

where

$$\sigma_{\max} = \max \left\{ \sqrt{\sum x_i^2}, \sqrt{\sum y_i^2}, \sqrt{\sum z_i^2} \right\}.$$

**5.1.2 Linear Fits (Lines and Planes)** For lines and planes, orthogonal distance regression can be formulated as a linear algebra problem. Within ATEP-CMS, the reference fits are computed using the singular value decomposition of the data matrix after translating the centroid to the origin. The properties of the singular value decomposition are well understood, and the magnitude of the error in the direction vector can be bounded using standard techniques [20] as follows.

We consider first the case of line fits. The direction of the line is the right eigenvector corresponding to the largest singular value of the data matrix. The singular value decomposition is computed using C-language double (64 bit IEEE real format) arithmetic, for which the unit roundoff is  $\varepsilon_{64} \approx 2.22 \times 10^{-16}$ . Call the data matrix  $D$  and denote the exact solution by  $\mathbf{Q}^*$ . It can be shown that the computed solution,  $\mathbf{Q}$ , is (within  $\varepsilon_{64}$ ) the exact solution to a matrix  $D + E$  where  $\|E\|_2 \leq \varepsilon_{64} \|D\|_2$ . (Here,  $\|\cdot\|_2$  denotes the vector or matrix 2-norm.) From this, it can be shown that [20], as long as the largest singular value is well separated<sup>9</sup> from the next largest singular value in comparison to  $\|E\|_2$ , the error in the computed solution is bounded by

$$\|\mathbf{Q} - \mathbf{Q}^*\|_2 \leq 4\varepsilon_{64} \frac{\lambda_1^2}{|\lambda_1^2 - \lambda_2^2|} \quad (\text{for lines}),$$

<sup>9</sup> The notion of “well separated” can be made precise. Essentially, it means that the data has one dominant direction of scatter (so a line fit makes sense).

where  $\lambda_1$  is the largest singular value and  $\lambda_2$  is the next-largest singular value. (This bound is not the tightest possible. For instance, as  $\lambda_2 \rightarrow \lambda_1$ , the bound can far exceed the theoretical worst-case difference between  $\mathbf{Q}$  and  $\mathbf{Q}^*$  of 2. In such situations, however, the singular values are not well separated. Within ATEP-CMS, the error bound is never allowed to exceed 2.)

Similar results can be obtained for plane fits. In this case, however, the normal to the plane is the right eigenvector corresponding to the *smallest* singular value,  $\lambda_3$ . The corresponding error bound is

$$\|\mathbf{Q} - \mathbf{Q}^*\|_2 \leq 4 \varepsilon_{64} \frac{\lambda_3^2}{|\lambda_2^2 - \lambda_3^2|} \quad (\text{for planes}).$$

We have here developed overall bounds for the errors in the eigenvectors. To convert these to standard uncertainties, we assume the errors are uniformly distributed within the bounds. The resulting standard deviation, denoted  $u_D$ , is given by

$$u_D = \begin{cases} \frac{4 \varepsilon_{64} \lambda_1^2}{\sqrt{3} |\lambda_1^2 - \lambda_2^2|} & \text{(lines)} \\ \frac{4 \varepsilon_{64} \lambda_3^2}{\sqrt{3} |\lambda_2^2 - \lambda_3^2|} & \text{(planes).} \end{cases}$$

These results are quite conservative. In general any one eigenvector will be more sensitive in one direction than another to perturbations in the data. This behavior could, in principle, be used to develop tighter bounds on the numerical errors propagated through the difference computations. Such an analysis will be the subject of future work.

**5.1.3 Nonlinear fits** For geometries other than line and plane (circle, sphere, cylinder, cone, and torus), orthogonal distance regression is a nonlinear least-squares optimization problem. (Circle fitting uses plane fitting to reduce the problem to two dimensions.) The algorithm currently used in ATEP-CMS to compute the reference fits is a modified Levenberg-Marquardt algorithm proposed by Nash [21].

The ATEP-CMS algorithm starts with an initial guess and iteratively solves the normal equations for a linear approximation to the nonlinear objective function, where the solution is constrained to lie within a “trust region” of the current best guess. The iterations terminate when the computed solution to the normal equations changes the solution by less than a convergence factor set by the tester. (Typically, the convergence factor used in ATEP-CMS is  $10^{-15}$ .)

The uncertainty of the reference fit is a combination of two factors. First, the termination logic introduces an

uncertainty that is bounded by the convergence factor setting. Second, the solution to the normal equations at the terminating iteration is subject to numerical round-off effects.

The convergence factor introduces an uncertainty in each element of the computed solution. If the calculated solution is a  $k$ -dimensional vector  $\mathbf{F}$ , if  $\mathbf{F}^*$  is the exact solution, and if  $C$  is the convergence factor, then the resulting error due to  $C$  alone is bounded by

$$\|\mathbf{F} - \mathbf{F}^*\|_2 \leq C \sqrt{k}.$$

We next deal with the numerical inaccuracies. The fits are calculated using the Choleski decomposition of the normal equations in C-language double arithmetic, so the error in the solution due to numerical roundoff effects alone can be bounded by

$$\|\mathbf{F} - \mathbf{F}^*\|_2 \leq \varepsilon_{64} \kappa(A) \|\mathbf{F}\|_2$$

where  $\kappa(A)$  is the condition number of the normal equation matrix  $A$  used for the last iteration of the Levenberg-Marquardt routine, and  $\varepsilon_{64} \approx 2.22 \times 10^{-16}$  as before [20].

To obtain an overall uncertainty, we must convert these bounds into standard deviations. We assume that the numerical errors and the convergence factor errors are independent and uniformly distributed between the bounds. Denote by  $u_F$  the standard uncertainty for nonlinear fits. Then

$$u_F = \sqrt{\frac{C^2 k}{3} + \frac{\varepsilon_{64}^2 \kappa^2(A) \|\mathbf{F}\|_2^2}{3}}.$$

As with linear fits, the bounds developed here bracket the overall behavior of the fit parameter vector. But as with the singular value decomposition, the solution to the nonlinear problem will in general exhibit greater sensitivity and correlations for some parameters than for others. Thus, it may be possible to obtain tighter uncertainty bounds than those developed here. One problem is that, for geometries other than the sphere, the parameterization used for fitting must somehow represent the orientation of the element. Unfortunately, there is no parameterization of orientations that is free of singularities, and there is always the possibility that the Jacobian of the residuals is rank deficient. This makes traditional approaches using, for instance, the condition number of the asymptotic covariance matrix at the computed solution problematic [22]. As with the singular value decomposition, future work will address tightening the bounds on the fit uncertainties.

## 5.2 Propagation to Difference Parameters

This section discusses how the uncertainties of the computed reference fits propagate to the difference parameters for individual fits. As with Sec. 5.1, the results of this section are specific to the particular algorithms currently being used within ATEP-CMS.

The previous section identified several sources of standard uncertainty for the reference fits. We have designated these as follows:

$u_0$ —the standard deviation of the centroid coordinates

$u_D$ —the standard deviation of the error in direction for line and plane fits

$u_F$ —the standard deviation of the error norm of the representation parameter vector for nonlinear fits.

We will separately propagate the standard uncertainties represented by each of these standard deviations through the comparison algorithms and combine the resulting standard uncertainties.

The calculations are fairly straightforward, but tedious. Therefore, we will show details only for the case of the uncertainties of line fits. For other geometry types, we will just present the final results, since the method is the same.

**5.2.1 Line** Line differences are defined by two characteristics: the angle between the test and reference lines, and the distance of an endpoint of the test line segment to the reference line. For the angle, we proceed as follows. As in Sec. 5.1.2, let  $\mathbf{Q}$  be the direction of the computed reference line and  $\mathbf{Q}^*$  the direction of the true reference line. Also, let  $\mathbf{Q}_T$  be the direction vector of the test fit,  $\alpha$  be the angle between  $\mathbf{Q}$  and  $\mathbf{Q}_T$ , and  $\alpha^*$  be the angle between  $\mathbf{Q}_T$  and  $\mathbf{Q}^*$ . Since all the direction vectors are unit vectors, the sine of the angle between any two of the vectors is the magnitude of their vector cross-product. We then have:

$$\begin{aligned} \sin(\alpha) &= \|\mathbf{Q}_T \times \mathbf{Q}\|_2 \\ &= \|\mathbf{Q}_T \times (\mathbf{Q}^* + (\mathbf{Q} - \mathbf{Q}^*))\|_2 \\ &= \|\mathbf{Q}_T \times \mathbf{Q}^* + \mathbf{Q}_T \times (\mathbf{Q} - \mathbf{Q}^*)\|_2 \\ &\leq \|\mathbf{Q}_T \times \mathbf{Q}^*\|_2 + \|\mathbf{Q}_T \times (\mathbf{Q} - \mathbf{Q}^*)\|_2 \\ &\leq \sin(\alpha^*) + \|\mathbf{Q} - \mathbf{Q}^*\|_2. \end{aligned}$$

If we start with  $\sin(\alpha^*) = \|\mathbf{Q}_T \times \mathbf{Q}^*\|_2$  and follow a similar sequence, we find that  $\sin(\alpha^*) \leq \sin(\alpha) + \|\mathbf{Q} - \mathbf{Q}^*\|_2$ . Assuming that  $\alpha$  and  $\alpha^*$  are small (i.e., that the

test and reference results are close),  $\sin(\alpha) \approx \alpha$  and  $\sin(\alpha^*) \approx \alpha^*$ . Thus,  $|\alpha - \alpha^*| \leq \|\mathbf{Q} - \mathbf{Q}^*\|_2$ . Using the results of Sec. 5.1.2,  $\|\mathbf{Q} - \mathbf{Q}^*\|_2$  is the random variable corresponding to a standard deviation of  $u_D$ . We then have

$$u_\alpha = u_D.$$

We now consider the distance from any point  $b$  to the reference line. Call this distance  $d$ . The reference line is located by the centroid of the data,  $\mathbf{q}_0$ , so  $d = \|(\mathbf{q} - \mathbf{q}_0) \times \mathbf{Q}\|_2$ . Two components of the fit uncertainty affect the uncertainty of  $d$ : the uncertainty of  $\mathbf{Q}$  and the uncertainty of  $\mathbf{q}_0$ . We treat these as independent sources of uncertainty and combine them in quadrature. With regard to  $\mathbf{Q}$ , we proceed as we did with  $\alpha$  and find that the uncertainty in  $d$  due to the uncertainty in  $\mathbf{Q}$  is bounded by

$$\|\mathbf{q} - \mathbf{q}_0\|_2 \|\mathbf{Q} - \mathbf{Q}^*\|_2.$$

The standard uncertainty in  $d$  due to the standard uncertainty in  $\mathbf{q}_0$  is  $u_0$ . Thus, the standard uncertainty of the separation parameter for lines is

$$u_d = \sqrt{u_0^2 + \|\mathbf{q} - \mathbf{q}_0\|_2^2 u_D^2},$$

where  $\mathbf{q}$  is now the endpoint on the test line segment furthest from the reference line.

**5.2.2 Circle** The standard uncertainties in the difference parameters for circles are as follows:

- distance between centers:  $u_c = \sqrt{u_0^2 + u_F^2}$
- angle between planes (due to the plane fits):  $u_\alpha = u_D$
- difference in radii:  $u_r = u_F$ .

**5.2.3 Plane** The standard uncertainties in the difference parameters for planes are as follows:

- angle between planes:  $u_\alpha = u_D$
- plane separation:  $u_d = \sqrt{u_0^2 + \|\mathbf{q} - \mathbf{q}_r\|_2^2 u_F^2}$ , where  $\mathbf{q}$  is the point on the test plane furthest from the reference plane and  $\mathbf{q}_r$  is the centroid of the data.

**5.2.4 Sphere** The standard uncertainties in the difference parameters for spheres are as follows:

- distance between centers:  $u_c = \sqrt{u_0^2 + u_F^2}$
- difference in radii:  $u_r = u_F$ .

**5.2.5 Cylinder** The standard uncertainties in the difference parameters for cylinders are as follows:

- angle between axes:  $u_\alpha = u_F$
- axis separation:  $u_d = \sqrt{u_0^2 + \|\mathbf{q} - \mathbf{q}_r\|_2^2 u_F^2}$  where  $\mathbf{q}$  is the endpoint of the test fit axis furthest from the reference fit axis and  $\mathbf{q}_r$  is the point used to locate the reference fit axis
- difference in radii:  $u_r = u_F$

**5.2.6 Cone** The standard uncertainties in the difference parameters for cones are as follows:

- angle between axes:  $u_\alpha = u_F$
- axis separation:  $u_d = \sqrt{u_0^2 + \|\mathbf{q} - \mathbf{q}_r\|_2^2 u_F^2}$ , where  $\mathbf{q}$  is the endpoint of the test fit axis furthest from the reference fit axis and  $\mathbf{q}_r$  is the point used to locate the reference fit axis
- difference in axial location:  $u_l = \sqrt{u_0^2 + u_F^2}$
- difference in half-angle:  $u_h = u_F$

**5.2.7 Torus** The standard uncertainties in the difference parameters for tori are as follows:

- angle between planes:  $u_\alpha = u_F$
- center separation:  $u_c = \sqrt{u_0^2 + u_F^2}$
- difference in major radii:  $u_M = u_F$
- difference in minor radii:  $u_m = u_F$

### 5.3 Uncertainty of the Estimated Performance Parameters

The results of Sec. 5.2 provide us with a means of estimating the uncertainties, expressed as standard deviations, of individual difference parameters  $p_i$  for individual data sets. Using this, we can now estimate the uncertainty of the software performance measures for each geometry type.

Recall (from Sec. 4.2) that for a particular performance characteristic, we are estimating the performance measure by the observed root-mean-square value of the difference parameters:

$$\hat{\gamma}_P = \sqrt{\frac{1}{n} \sum p_i^2}.$$

Furthermore, we have an estimate, denoted by  $u_i$ , for the standard deviation of each  $p_i$  attributable to the uncertainty of the reference fit.

Our objective is to estimate the variance of  $\hat{\gamma}_P$ . To that end, we will use the following identity:

$$V(\hat{\gamma}_P) = E_A [V(\hat{\gamma}_P|A)] + V_A [E(\hat{\gamma}_P|A)],$$

where  $A$  is a random event,  $E(\hat{\gamma}_P|A)$  and  $V(\hat{\gamma}_P|A)$  are,

respectively, the expected value and variance of  $\hat{\gamma}_P$  conditional on event  $A$ , and  $E_A[\cdot]$  and  $V_A[\cdot]$  are, respectively, the expected value and variance over all events  $A$ . This identity, which follows from the theorem of total probability, helps us estimate the variance of  $\hat{\gamma}_P$  by using the observed sample of performance for the data sets we used.

Let  $A$  be the event that the sample of observations is the one we indeed observed. (By conditioning on  $A$  we are treating the sample as fixed and not random.) We can interpret the two terms of the variance as follows. The first term,  $E_A[V(\hat{\gamma}_P|A)]$  represents the component of standard uncertainty due to the uncertainty of the reference fits. We will denote this quantity by  $u_R^2$ , meaning “square of the standard uncertainty due to the reference.” The second term,  $V_A[E(\hat{\gamma}_P|A)]$ , represents the component of standard uncertainty due to the variation in the sampling. We will denote this by  $u_S^2$ , meaning “square of the standard uncertainty due to sampling.” We now develop estimates for the two components of uncertainty, starting with  $u_R$ .

**5.3.1 Uncertainty Due to the Reference** To estimate  $u_R$ , we must first estimate  $V(\hat{\gamma}_P|A)$ . Considering  $\hat{\gamma}_P$  as a function of the  $p_i$ , we have, using the law of propagation of uncertainty

$$V(\hat{\gamma}_P|A) \approx \sum \left[ \frac{\partial \hat{\gamma}_P}{\partial p_i} \right]^2 u_i^2 = \frac{\sum p_i^2 u_i^2}{n \sum p_i^2}.$$

(If all the  $p_i$  are zero (as might happen, for example, if the NIST algorithms are tested against themselves), then this estimate is indeterminate. However, whenever  $p_i$  is small in relation to  $u_i$ , we can improve the estimate by evaluating the partial derivatives in the propagation formula at the expected value of  $|p_i|$  instead of at the expected value of  $p_i$ . To do this, we assume that  $p_i$  is distributed uniformly about the observed value within  $\pm \sqrt{3}u_i$ . Then, whenever  $|p_i| < \sqrt{3}u_i$  the mean should be modified to include the “folding back” at zero of the distribution of  $|p_i|$ . The result is

$$V(\hat{\gamma}_P|A) \approx \frac{\sum \tilde{p}_i^2 u_i^2}{n \sum \tilde{p}_i^2}$$

where

$$\tilde{p}_i = \begin{cases} p_i & \text{if } p_i^2 \geq 3u_i^2 \\ \frac{p_i^2 + 3u_i^2}{2\sqrt{3}u_i} & \text{if } p_i^2 < 3u_i^2 \end{cases}$$

This approximation to  $V(\hat{\gamma}_P|A)$  is used in ATEP-CMS as an estimate of  $E_A[V(\hat{\gamma}_P|A)]$

$$\hat{u}_r^2 = \frac{\sum \tilde{p}_i^2 u_i^2}{n \sum \tilde{p}_i^2}$$

(If all the  $\tilde{p}_i$  are zero—that is,  $u_i = 0$  whenever  $p_i = 0$ —then we set  $\hat{u}_R = 0$ .)

The component of standard uncertainty due to the reference is estimated from numerical analysis considerations and, in the language of the guidelines on uncertainty [11, 10], is a Type B standard uncertainty.

### 5.3.2 Uncertainty Due to Sampling Variation

We now turn to  $u_s^2$ , the component of standard uncertainty due to sampling. To estimate  $u_s^2$ , we need to estimate  $E(\hat{y}_p|A)$ . We do this by simply using a zero-order expansion of  $\hat{y}_p$  about the  $p_i$ , giving  $E(\hat{y}_p|A) \approx \hat{y}_p$ , and we are left with estimating  $V_A(\hat{y}_p)$ . If  $\hat{y}_p = 0$ , we estimate  $V_A(\hat{y}_p) = 0$ . Otherwise we proceed as follows.

By the law of propagation of uncertainty,

$$V_A(\hat{y}) \approx \frac{1}{4\hat{y}_p^2} V_A(\hat{y}_p^2) = \frac{1}{4n^2\hat{y}_p^2} \sum V_A(p_i^2).$$

For convenience, we denote by  $\sigma_A^2$  the mean variance of the  $p_i^2$  over all samples

$$\sigma_A^2 = \frac{1}{n} \sum V_A(p_i^2).$$

Since  $\hat{y}_p$  is estimated from the sample, an unbiased estimate for  $\sigma_A^2$  is:

$$\hat{\sigma}_A^2 = \frac{1}{n-1} \sum (p_i^2 - \hat{y}_p^2)^2.$$

We then have

$$\hat{u}_s^2 = \frac{\sum (p_i^2 - \hat{y}_p^2)^2}{4n(n-1)\hat{y}_p^2} = \frac{\sum (p_i^2 - \hat{y}_p^2)^2}{4(n-1) \sum p_i^2},$$

with  $\hat{u}_s = 0$  if all of the  $p_i = 0$ .

Since the component of uncertainty due to sampling is estimated from a statistical analysis of data, it is a Type A standard uncertainty, with  $n-1$  degrees of freedom.

**5.3.3 Combined Standard Uncertainty** Putting the two terms of the variance identity together, the combined standard uncertainty of the estimated performance parameter  $\hat{y}_p$  is the positive square root of the estimated variance of  $\hat{y}_p$

$$V(\hat{y}_p) = u_R^2 + u_s^2 \approx \frac{\sum \tilde{p}_i^2 u_i^2}{n \sum \tilde{p}_i^2} + \frac{\sum (p_i^2 - \hat{y}_p^2)^2}{4(n-1) \sum p_i^2}.$$

Following NIST convention, the expanded uncertainty reported by ATEP-CMS is twice the combined standard uncertainty (i.e., a coverage factor of  $k = 2$ ).

## 6. Summary

This paper has described how NIST evaluates the performance of geometric fitting software used for inspection. The NIST service, ATEP-CMS, is the only known test that provides quantitative measures of performance, complete with statements of uncertainty in accordance with international standards.

ATEP-CMS is something new in the way of calibration services. It is the only test of software offered by NIST in the field of dimensional metrology. It has the status of a Special Test, rather than a Calibration Service, because it is an experiment for us. The statements of uncertainty, in particular, are unsupported by any historical data. We believe, however, that they are fundamentally sound. We have tested some of our capabilities by running our software on both a personal computer and on NIST's supercomputer. These limited results support the validity of our approach.

This paper has focused on the performance measures used in ATEP-CMS; we have not discussed testing procedures. However, one key aspect of those procedures bears discussion: the selection of data sets to be used for the test. We use the collection of data sets as if the average performance of the software over those problems represents the average performance of the software when it is used in production. To help ensure this, we use a stratified sampling approach in designing data sets for a test. Through judgment, we define ranges of parameters for inspection problems, including ideal geometry, form errors, surface sampling plans, and point measurement errors. Within each range, we select a representative sampling of data sets. We believe that this mixed approach improves the quality of the test.

We expect to refine our procedures as we gain experience in testing. We also plan to extend the scope of our testing services beyond orthogonal distance regression to include other fitting methods and other common CMS software functions [23].

The methods described in this paper demonstrate that classical concepts of metrology can be used to assess the performance of software, when that software forms part of a measurement system. Moreover, it seems that performance testing like that offered by ATEP-CMS is a requirement if CMS measurements are to be traceable to accepted standards of length.

## Acknowledgments

We wish to thank several people who have contributed to the results discussed in this paper. Our many discussions with Keith Eberhardt helped resolve several statistical issues we faced in developing the performance measures. We would also like to thank Dan Lozier for his thoughtful review of an earlier draft of this paper. Cathleen Diaz has been the driving force that has turned ATEP-CMS from an idea into reality. David Rosenfeld's expert programming and testing efforts have ensured that our testing system is up to snuff and capable of doing the work described herein. Ann Nishioka worked through the details of implementing and testing the code to estimate the uncertainty of NIST fitting algorithm results. Finally, we wish to acknowledge the work of the ASME B89.4.10 Working Group in developing the draft standard which our work supports.

## 7. References

- [1] C. Diaz, Concept for an Algorithm Testing and Evaluation Program at NIST, NISTIR 5366, National Institute of Standards and Technology, Gaithersburg, MD (January 1994).
- [2] T. H. Hopp, Computational Metrology, Manufacturing Review **6**(4), 295–304 (1983).
- [3] C. Porta and F. Wäldele, Testing of Three Coordinate Measuring Machine Evaluation Algorithms, Physicalisch-Technische Bundesanstalt, Report EUR 10909 EN, Directorate—General Science, Research and Development, Commission of the European Communities, Brussels, Belgium (1986).
- [4] A. Weckenmann and M. Heurichowski, Problems with software for running coordinate measuring machines—the use of virtual volumetric standards, Precision Engineering **7**(2), 87–91 (1985).
- [5] R. Walker, GIDEP Alert No. X1-A-88-01, Government-Industry Data Exchange Program, Washington, DC (August 22, 1988).
- [6] ASME, Methods for Performance Evaluation of Coordinate Measuring System Software, ASME B89.4.10-199x, American Society of Mechanical Engineers, New York, NY, in preparation.
- [7] R. W. Farebrother, Mechanical Representations of the  $L_1$  and  $L_2$  Estimation Problems, in Statistical Data Analysis Based on the  $L_1$ -Norm and Related Methods, Y. Dodge (ed.), Elsevier, New York, NY (1987) pp. 455–464.
- [8] W. Gander, G. H. Golub, and R. Strebel, Fitting of Circles and Ellipses: Least Squares Solution, Report 217, Departement Informatik, ETH, Zürich (June 1994).
- [9] C. Diaz, Algorithm Testing and Evaluation Program for Coordinate Measuring Systems: Testing Methods, NISTIR 5686, National Institute of Standards and Technology, Gaithersburg, MD (1995).
- [10] B. N. Taylor and C. E. Kuyatt, Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results, NIST Technical Note 1297, 1994 Edition, National Institute of Standards and Technology, Gaithersburg, MD (1994).
- [11] ISO, Guide to the Expression of Uncertainty in Measurement, International Organization for Standardization, Geneva, Switzerland (1993).
- [12] P. T. Boggs, R. H. Byrd, and R. B. Schnabel, A Stable and Efficient Algorithm for Nonlinear Orthogonal Distance Regression, SIAM J. Sci. Stat. Comp. **8**(6), 1052–1078 (1987).
- [13] M. G. Cox and A. B. Forbes, Strategies for Testing Form Assessment Software, NPL Report DITC 211/92, National Physical Laboratory, Teddington, Middlesex, UK, (December 1992).
- [14] ASME, Dimensioning and Tolerancing, ASME Y14.5M-1982, American Society of Mechanical Engineers, New York, NY (1982).
- [15] M. H. Hansen, W. N. Hurwitz, and W. G. Madow, Sample Survey Methods and Theory, Vol. 1, Wiley, New York, NY (1953).
- [16] M. E. A. Algeo and C. Diaz, A Process for Selecting Standard Reference Algorithms for Evaluating Coordinate Measurement Software, NISTIR 5374, National Institute of Standards and Technology, Gaithersburg, MD (January 1994).
- [17] D. Goldberg, What Every Computer Scientist Should Know About Floating-Point Arithmetic, Computing Surveys **23**(1), 5–48 (1991).
- [18] J. H. Wilkinson, Rounding Errors in Algebraic Processes, Prentice Hall, Englewood Cliffs, NJ (1963).
- [19] P. L. Meyer, Introductory Probability and Statistical Applications, Addison-Wesley, Reading, MA (1970).
- [20] G. H. Golub and C. F. van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, MD (1983).
- [21] J. C. Nash, Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation, Adam Hilger, Ltd., Bristol, England (1979).
- [22] P. T. Boggs and J. R. Donaldson, The Computation and Use of the Asymptotic Covariance Matrix for Measurement Error Models, NISTIR 89-4102, National Institute of Standards and Technology, Gaithersburg, MD (1989).
- [23] S. C. Feng and T. H. Hopp, A Review of Current Geometric Tolerancing Theories and Inspection Data Analysis Algorithms, NISTIR 4509, National Institute of Standards and Technology, Gaithersburg, MD (February 1991).

**About the authors:** *Theodore H. Hopp is a computer scientist in the Manufacturing Systems Integration Division of the NIST Manufacturing Engineering Laboratory. Mark S. Levenson is a mathematical statistician in the Statistical Engineering Division of the NIST Computing and Applied Mathematics Laboratory. The National Institute of Standards and Technology is an agency of the Technology Administration, U.S. Department of Commerce.*